Software Engineering Institute

Survivability Assurance for System of Systems

Robert J. Ellison
John Goodenough
Charles Weinstock
Carol Woody

May 2008

TECHNICAL REPORT
CMU/SEI-2008-TR-008
ESC-TR-2008-008

Networked Systems Survivability

Carnegie Mellon

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

Acknowledgments

This work is the result of many incremental pieces and could not have been accomplished without the support, involvement, and encouragement of each contributor.

The core approach for the Survivability Analysis Framework (SAF) was conceived in support of a project for Office of the Under Secretary of Defense for Acquisition, Technology and Logistics (OUSD [AT&L]) to study how survivability of critical multi-service capabilities might be impacted by planned increased interoperability. This framework was initially developed to analyze mission survivability of the Joint Battle Management Command and Control (JBMC2) business processes (also known as mission threads) as the U.S. Department of Defense (DoD) pushes toward increased interoperability with the envisioned Global Information Grid (GIG). Limitations in available analysis techniques required those of us working on that project to build a new way to characterize and analyze business processes. Researchers from the Software Engineering Institute's Dynamic Systems Program worked with researchers from SEI's Networked Survivable Systems (NSS) Program to assemble this initial effort.

Expansion and refinement of the SAF continued through projects supported by the U.S. Air Force's Electronic Systems Center (ESC) Cryptologic Systems Group (CPSG) to identify ways of connecting information assurance acquisition decisions to mission critical needs. In partnership with Robert Flores, lead security engineer at CPSG/NIS, GIG IA Solutions Division, and David Eyestone, CPSG/NIS, the NSS team was able to strengthen the structure of the framework to better support ways of connecting information assurance needs to mission drivers.

Through work funded by Edgar Dalrymple, associate director for software and distributed systems, PM Future Combat Systems (Brigade Combat Team), the potential for building an assurance case with information about business processes from the SAF was identified. Preliminary usage of SAF with reliability and security assurance cases for FCS provided an opportunity to experiment with blending the analysis techniques.

A "seed grant" from Carnegie Mellon Cylab for academic year 2006-2007 provided the support for the authors to develop publicly available materials describing the framework and its use in assembling an example assurance case.

Executive Summary

The growing need for interconnected and often global operations means business processes are structured to include many organizations, and technology support is multi-system. Few analysis techniques provide a way to characterize beyond the limitations of the single system and many are also limited to a finite set of stakeholders. This results in organizations failing to identify and address the growing challenges of systems of systems [Maier 1998]. Section 1 of this report is focused on helping the reader understand the complexity and challenges of systems of systems.

The Survivability Analysis Framework (SAF), a structured view of people, process, and technology, was developed to help organizations characterize the complexity of multi-system and multi-organizational business processes. Survivability gaps arise when assumptions and decisions within one organizational area are inconsistent with those of another, resulting in differences and conflicts among the systems developed and used to support each organizational area. SAF provides a structure for capturing information about a business process so that gaps are readily identifiable. The SAF is designed to address the following:

- identify potential problems with existing or near-term interoperations among components within today's network environments
- highlight the impact on survivability as constrained interoperation moves to more dynamic connectivity
- increase our assurance that the business process can survive in the presence of stress and possible failure

Section 2 of this report describes, through the use of a medical business process example, the steps required to apply SAF and the resulting artifacts. Failure analysis opportunities are introduced using the artifacts constructed in the medical example. Much of the information needed to assemble this view is scattered among a range of stakeholders and must be gathered through documents and workshops. Pilot usage of SAF has shown that most characterizations of business processes are idealized, providing insight into how they should work without consideration for what is actually in place. When technology is developed to only address ideal usage, actual operational usage is poorly supported.

The third section of this report introduces the *assurance case*, a method for documenting justified confidence that survivability has been adequately addressed. Much like a legal case presented in a courtroom, an assurance case is a comprehensive presentation of evidence with argumentation linking the evidence with claims that certain properties have been satisfied. With the construction of a structured view of a business process using SAF, a great deal of the evidence needed to support the claims of an assurance case can become visible. The steps needed to assemble the assurance case are described using the medical business process example developed in Section 2.

By combining these two analysis techniques, the strengths and gaps for the survivability of a business process can be described in a graphical and visually compelling form that management, architects, system engineers, software engineers, and users can share.

Abstract

Complexity and change pervade today's organizations. Organizational and technology components that must work together may be created, managed, and maintained by different entities. Net-centric operations and service-oriented architectures will push this trend further, increasing the layers of people, processes, and systems. Existing analysis mechanisms do not provide a way to (1) focus on challenges arising from integrating multiple systems, (2) consider architecture tradeoffs carrying impacts beyond a single system, and (3) consider the linkage of technology to critical organizational functions. In response, a team at the Software Engineering Institute (SEI) built an analysis framework to evaluate the quality of the linkage among roles, dependencies, constraints, and risks for critical technology capabilities in the face of change.

The Survivability Analysis Framework (SAF), a structured view of people, process, and technology, was developed to help organizations analyze and understand stresses and gaps to survivability for operational and proposed business processes. The SAF is designed to

- identify potential problems with existing or near-term interoperations among components within today's network environments
- highlight the impact on survivability as constrained interoperation moves to more dynamic connectivity
- increase assurance that mission threads can survive in the presence of stress and possible failure

1 Introduction

Complexity and change are pervasive in the operational environments of today's organizations. Organizational and technology components that must work together may be created, managed, and maintained by different entities around the globe. Net-centric operations and service-oriented architectures will push this trend further, increasing the layers of people, processes, and systems that must work together for successful completion of a business process. Existing analysis mechanisms do not provide a way to (1) focus on challenges that arise from integrating multiple systems, (2) consider architecture tradeoffs that carry impacts beyond a single system, and (3) consider the linkage of technology to critical organizational functions. In response, a team at the Carnegie Mellon University Software Engineering Institute (SEI) built an analysis framework to evaluate the quality of the linkage among roles, dependencies, constraints, and risks for critical technology capabilities in the face of change.

Section 1 of this report is focused on helping the reader understand the complexity and challenges of systems of systems. Section 2 of this report describes, through the use of a medical business process example, the steps required to apply the Survivability Analysis Framework (SAF) and the resulting artifacts. Failure analysis opportunities are introduced using the artifacts constructed in the medical example. The third section of this report introduces the *assurance case*, a method for documenting justified confidence that survivability has been adequately addressed. By combining these two analysis techniques, the strengths and gaps for the survivability of a business process can be described in a graphical and visually compelling form.

1.1 THE NEED FOR ANALYSIS OF SYSTEMS OF SYSTEMS

Increasingly, business work processes require integration across multiple systems, essentially an enterprise system of systems [Maier 1998]. For example, a work process that supports a just-in-time supply chain for manufacturing can involve multiple organizations. It is clear that the move toward systems of systems (SoS) is increasing business and government use of software at unprecedented levels of scale and complexity. Software is, indeed, the mechanism that enables systems of systems to function. Add to this the move toward decentralization and the pace at which business and mission requirements change, and a great deal of uncertainty results regarding both the configuration of the SoS at any given time and the behavior that can be expected by its constituents. Greater scale, uncertainty, and complexity bring with them a rapidly growing set of failure modes. Hence, requirements for software assurance and other quality attributes related to software dependability and supportability need a strong emphasis. Historically, except for safety-critical systems and systems controlling financial transactions, efforts to build in these quality attributes have had much lower priority than efforts to develop functionality. This must change, and will only change when the acquirer provides incentives for performance with respect to assurance and quality requirements, not just cost, schedule, and the delivered domain functionality.

Technologies such as Web services make it easier to assemble systems, but ease of assembly may only increase the risk of deploying systems whose behavior is not predictable. Fairly simple computing architectures that could be understood and their behavior characterized have been replaced by distributed, interconnected, and interdependent networks. The theme of a 2007 *New York Times* article is captured in a quote by Peter Neumann: "We don't need hackers to break the systems because they're falling apart by themselves" [Schwartz 2007]. For example, 17,000 international travelers flying into Los Angeles International Airport were stranded on planes for hours one day in mid-August 2007 after U.S. Customs and Border Protection Agency computers went down and stayed down for nine hours. The power grid failure in the northeastern United States and Canada in the summer of 2003 is another recent example of the effects of a system failure. Voting machine failures continue to be publicized. Customers of Skype, the Internet-based telephone company, encountered a 48-hour failure in August 2007.

The Los Angeles airport failure was traced to a malfunctioning network card on a desktop computer that slowed the network and set off a domino effect of failures on the customs network. The power grid failure was not caused by a single event but by a cascading set of failures, including a significant software failure affecting both the primary and backup operating structures. Aviel D. Rubin, a professor of computer science at Johns Hopkins University, noted that the assurance focus for voting machines might have been too much on hackers and not enough on accidental events that sometimes can cause the worst problems. The Skype failure was initiated by a deluge of login attempts by Skype users whose computers had restarted after downloading a monthly Microsoft security update. The logins overloaded the Skype network and revealed a bug in the Skype program that normally would have mitigated the excessive network load by reallocating computer resources [Schwartz 2007].

While the individuals interviewed for the *New York Times* article included a number of well-known computing security experts, the general observations focused more on the underlying complexity than on security.

- Most of the problems we have today have nothing to do with malice. Things break. Complex systems break in complex ways.
- Simpler systems could be understood and their behavior characterized, but greater complexity brings unintended consequences. Problems are increasingly difficult to identify and correct with the shift from "stovepipes" (stand-alone systems) to interdependent systems.
- Business usage requires change, but such change increases complexity by attempting to integrate incompatible computer networks or increasing the scale and scope of systems beyond the ability of the current capabilities to manage and sustain.

1.2 THE ASSURANCE PROBLEM IS HARD AND GETTING WORSE

Systems acquisition and development have historically focused on the development of a system that operated in a stand-alone fashion or had few interactions with other systems. The acquisition and development processes typically concentrated first on the functionality required and then on monitoring development and reviewing products to ensure the required functionality was provided. Requirements were assumed to be relatively static and development monitoring often concentrated on costs and schedule.

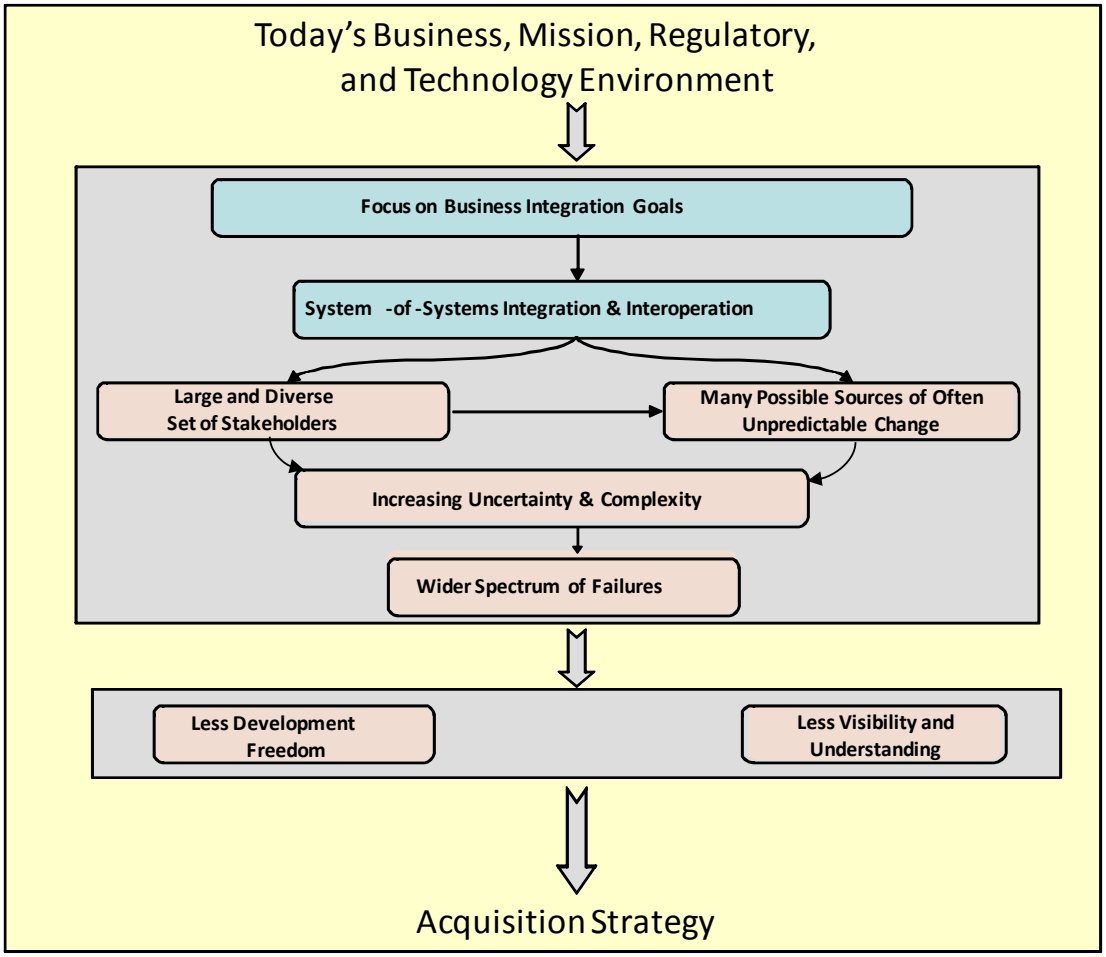


Figure 1: Development and Sustainment Context

As noted earlier, computing support for business work processes increasingly requires integration of multiple systems forming an SoS. The development of a single system that is a component of an SoS has to confront a number of development drivers as shown in Figure 1 that are not adequately addressed by the techniques appropriate for the development of a stand-alone system [Creel 2008].

The management of some drivers should among the success criteria for a design. Such drivers include

- large and diverse stakeholder community, which complicates requirements elicitation and tradeoff analysis
- potential for change from multiple directions at any time—from any system-of-systems constituent as well as from evolving business requirements
- wider spectrum of failures—users, operational, software, and systems—with causes or impacts beyond an individual system boundary

While drivers such as change-from-all-directions lead to challenging design problems, other drivers limit the design options or limit the analysis that can be done. Such drivers include

- limited development freedom—development choices are restricted by constraints associated with design tradeoffs and usage associated with existing systems
- limited or no knowledge of individual system structure and behavior and reduced or no visibility into runtime state

Large and Diverse Stakeholder Community

The evolution from stand-alone systems, to systems with a few known interfaces, to a system that is a component of an SoS increases the number and diversity of stakeholders. This increase, along with the potential volatility of the stakeholder community, results in conflicts of interest and understanding that may be difficult to resolve, creating technical and management problems.

Stakeholders are beginning to demand high levels of assurance for system qualities such as safety, security, reliability/availability/maintainability, performance, adaptability, interoperability, usability, and scalability. The stakeholders for a collection of systems may disagree on which quality attributes take precedence. In such an environment, a successful acquisition likely means that no one is completely satisfied, but everyone gets something they need and can use.

Unpredictable and Dynamic Change

Today, the focus is increasingly on the development of systems that are intended to function as constituents within a larger SoS context. Concurrently with development, business and mission needs for SoS constituents continue to evolve, and users expect an ability to adapt their systems accordingly. Along with providing new or modified capabilities, a system may need to communicate with other systems that were not identified up front. In such an environment, it becomes critical to specify requirements related to assurance goals and to build in the qualities needed to enable acceptable operation in the midst of a high degree of complexity and change.

In today's world, business and mission systems are expected to adapt to market changes and changes in the world environment, but often modifications have to be implemented where components such as commercial off-the-shelf (COTS) or legacy components cannot be easily changed. An individual system may support multiple work processes and hence must be responsive to changes in each. In a system-of-systems environment, with constituents independently managed and operated, adaptations one constituent makes to respond to change may result in unintended side effects, not only to the constituent system but to other systems as well. For example, the addition of devices such as cell phones or portable computers and the underlying software and networking technology can significantly affect risk mitigations and the system architecture.

Wider Spectrum of Failures

Technologies such as Web services make it easier to assemble systems, but ease of assembly may only increase the risk of deploying systems whose behavior is not predictable. Fairly simple computing architectures that could be understood have been replaced by distributed, interconnected, and interdependent networks. Business requirements increase the likelihood of failure by bringing together incompatible systems or by simply growing beyond the ability to manage change. As we

depend more on interdependent systems, failures are not only more likely but also harder to identify and fix.

An increasing number of failures are caused by unanticipated interactions between SoS constituents. Failures may be the result of discrepancies between the expected activity and the actual behavior that occurs normally in business processes. The overall success of a business process depends on how these discrepancies are dealt with by staff and supporting computing systems. Changes in business processes and systems often introduce these kinds of discrepancies.

Dealing with discrepancies becomes much more difficult as the number of participants—people and systems—increases. Each participant has to deal with multiple sources of discrepancies, and a single discrepancy can affect multiple participants. There is increased likelihood that a poorly managed discrepancy will result in additional discrepancies affecting additional participants. Failures are frequently the result of multiple, often individually manageable errors that collectively become overwhelming.

Limited Visibility and Understanding

The task of eliciting and communicating requirements, understanding system interfaces and usage patterns, and refining assurance strategies is never quite finished. A potentially unbounded stakeholder community, the number and diversity of components, systems, and services to be integrated, and evolution in both stakeholder needs and system configurations create unprecedented levels of uncertainty and complexity. It is virtually impossible to understand everything about a SoS, let alone influence all decisions made on behalf of its constituents.

System understanding is limited with COTS components, legacy systems, and with independently developed and managed systems. In addition, the complexity and evolving nature of a system of systems limit the ability to fully identify risks, understand the consequences, and analyze mitigations in advance of the start of development. Requirements will be incomplete. All parties to an acquisition are working with an incomplete understanding of the problem. As that understanding grows during development, all parties should be in a better position to understand the tradeoffs that may have to be made to resolve the problems. The acquisition consequences can be increased costs, delayed delivery, or the inability to satisfy a requirement.

Limited Development Freedom

Software is touted for its flexibility in terms of meeting requirements, but that flexibility is fully available only at the start of development and only to the extent that the environment allows. SoS development has a sustainment flavor. An acquisition for a SoS context rarely means clean slate development as the SoS is usually an existing operational system.

1.3 STRATEGY FOR A SOLUTION

The overall objectives for a solution include specifying a context that focuses analysis on the critical issues, building an understanding of the risks associated with this context, and finding sufficient common ground among the multiple perspectives to create an effective solution with known but acceptable limitations.

The development drivers discussed in section 1.2 suggest the criteria that are listed in Table 1, which help to analyze possible approaches.

Table 1: Evaluation Criteria

Development Drivers	Criteria for Approach Evaluation
Large and diverse stakeholder community	maintains traceability between technical decisions and business requirements provides a shared view that allows people with different perspectives to see the issues various stakeholders have captures success criteria reflecting the primary business drivers establishes a basis for resolving requirement conflicts provides a framework (context) for making tradeoffs among alternatives
Change from any direction	provides mechanisms for dealing with change from a variety of sources considers the effects of normal evolution of usage and technology in independently managed and developed systems
Wider spectrum of failures	provides mechanisms for managing the multitude of identified failure outcomes. defines acceptable risk for the diverse stakeholder community in the operational context shows what constitutes a “best effort” solution to a problem and hence demonstrates due diligence in identifying and mitigating risks
Limited visibility and knowledge	can be applied with incomplete information identifies the effects of new or changed information on the existing analysis

The SAF focuses on work processes (or what the DoD refers to as mission threads) that span computing systems. Evolving work processes are a significant driver of change and a source of often difficult integration problems. The focus on work processes enables better traceability from technological solutions to business priorities and provides a mechanism for developing a shared view among the many stakeholders. The SAF constructs an operational model for a work process that provides a context in which to reason about a wide spectrum of failures: technology, software, systems interoperability, operational, and human.

The limited visibility and knowledge in this context also means that the users of a computing service will not have immediate knowledge of a cause of a service failure and if that failure was maliciously induced. Failure analysis has to be general and not delegated to just security and safety analysis.

The SAF is introduced in Section 2 with an extended example.

The complexity associated with failure analysis can also affect testing by generating too many test cases. Safety cases have been extensively applied to justify safety claims about a system. A safety case can be generalized to an assurance case that can be applied to other quality attributes. Chapter 3 introduces assurance cases as a way to establish confidence that a system is sufficiently survivable by showing why significant survivability threats have been adequately mitigated or eliminated.

2 Building a Shared View of the Organization, People, Process, and Technology for Assurance

The Survivability Analysis Framework provides a mechanism for assembling the broad range of information that influences a business process in order to analyze it for quality and business survivability. Execution of a business process requires an extensive list of components working in harmony:

- hardware—servers, data storage devices, PCs, PDAs, routers, telephone switches, satellite relays, physical access controls, and similar devices
- software—operating systems for each hardware platform, configuration management, databases, firewalls, network protocols, packet switches, authentication packages, Web applications, local and remote procedures, and others
- people—organizational roles for data entry, inquiry, verification, audit, synthesis among multiple information sources, administration for technology components, authentication and authorization authorities, and similar roles
- policies and practices—certification and accreditation, third-party access management, outsourcing contracts, governance controls, and the like

From a pragmatic perspective, the responsibilities for quality and survivability are allocated across all of these components, which must function together to successfully achieve the work process's objective. The level of complexity is too much to validate without developing specific examples to characterize how all of the pieces should work together. From these examples, potential weak points can be identified; assumptions about the ways in which components will work together can be verified; and the criticality of each component to the success of the business process can be evaluated.

An important characteristic of business processes is that they are constantly changing. Software and hardware upgrades must be expected. In addition, an organization's needs are changing and the processes must adjust to these new requirements. SAF captures and analyzes the ways in which end-to-end business processes could be stressed and whether the stress-handling approaches applied within and among process steps are appropriate for successful process completion. Unlike existing analysis methods, SAF focuses on the challenge arising from integrating multiple business units and systems, considers tradeoffs beyond a single system, and considers the linkage of technology to critical organizational functions.

2.1 SURVIVABILITY ANALYSIS FRAMEWORK

The Survivability Analysis Framework[1] was developed to help organizations analyze and understand threats and gaps to survivability for complex operational business processes. In some domains, business processes are referred to as mission threads. We will use these terms interchangeably throughout this document. The growing need for interconnected, and often global operations, means business processes are less frequently bounded by a single system or contained within a single organizational unit. However, most widely used analysis techniques primarily focus on a single system controlled from a single organizational unit and miss the growing challenges of systems of systems. The need for a multi-system, multi-organizational view to define operational survivability in the face of ever-increasing complexity requires a new approach to characterizing the linkages between organizational mission and technology.

2.1.1 Mission Thread Steps and Step Interactions

Each critical step in a mission thread is tasked to fulfill some portion of mission thread functionality. This tasking represents a "contract" of interaction between the mission thread step and prior and subsequent steps. Preconditions establish the resources provided to the step. These preconditions may trigger the execution of the step (for example, data or a human command), or the process may be continually executed (such as a sensor). Each step will have outcomes (post conditions) that may interact with subsequent steps. However, the contract with prior and subsequent steps is not necessarily static and may have to be negotiated during execution to reflect the current situation. Even the identity of prior and subsequent steps may vary across executions of a business process.[2]

Environmental, data, process, and interaction limitations can lead to potential degradation of step actions. Each limitation represents a source of stress on the step and, consequently, on the business process. However, such stress does not necessarily cause failure. Steps can be designed to manage a range of stresses and still respond appropriately or degrade gracefully. Additionally, the failure of any specific step may not necessarily doom a process, because subsequent steps may continue to execute the thread.

Linkages among steps are driven by three primary components: people, resources (technology, systems, connectivity, policies, and the like), and actions. The behavior of the linkages coupled with the activities to be addressed in each step can also lead to stresses, and unmanaged stresses can potentially lead to interaction failure. Further incompatibilities arise if a step manages a stress in a manner that is not expected by subsequent steps. For example, consider a step that receives some data as input. If the value received by the step is out of the expected range, then the step can include actions to respond in a variety of ways. For instance, an action might substitute a default value in place of the out-of-range value. This substitution, however, may have dire consequences

---

[1] SAF was piloted for Joint Battle Mission Command and Control (JBMC2) in analysis of a Time Sensitive Targeting mission thread for the OUSD (AT&L). A second pilot analysis was completed for Time Sensitive Targeting information assurance for Electronic Systems Center, Cryptologic Systems Group, and Network Systems Division (ESC/CPSG NIS).

[2] Business processes are expected to be dynamic in content because each specific execution is unique.

if the decision to manage the stress by substituting a default value is inconsistent with the subsequent step's expectation for a highly accurate value.

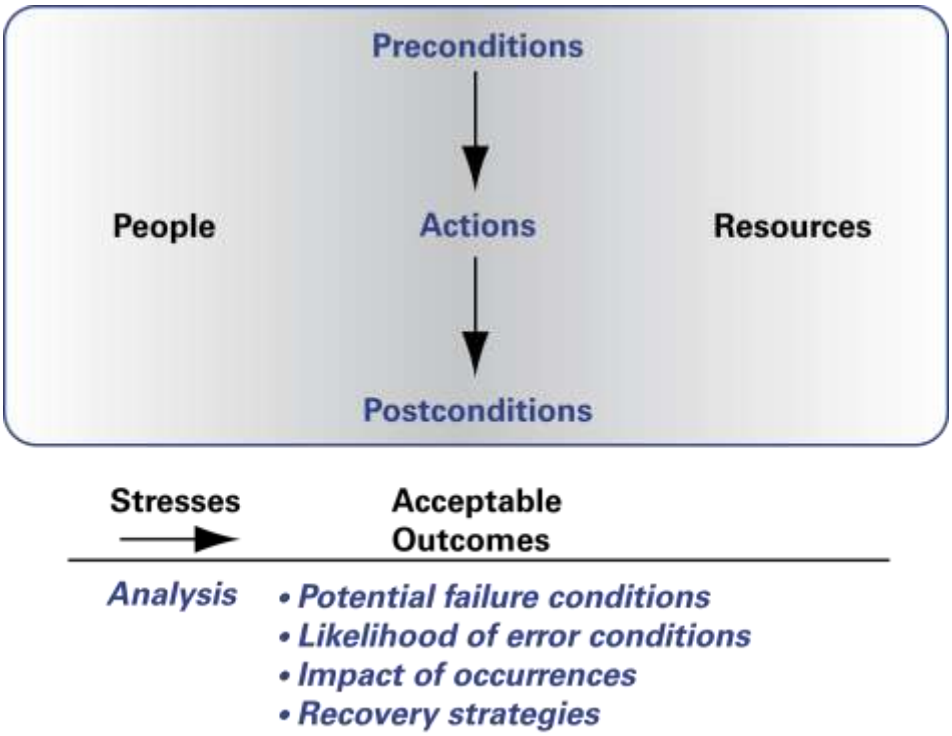


Figure 2. Survivability Analysis Framework

SAF characterizes the specific actions of each step in a business process and the linkages between each step. By evaluating successful business process completion and considering ways in which success could be jeopardized, SAF provides a mechanism to capture for analysis the stresses that may impact a business process. It also provides opportunities to analyze whether the stress-handling approaches adopted by a step are compatible with subsequent business process steps.

The SAF structure is applied using a process for characterizing a business process and expected quality, which is described in Section 2.1.2. A range of analysis mechanisms for evaluating the effectiveness of step interactions can use the SAF structure to identify potential failure conditions, impact of occurrences, and recovery strategies.

2.1.2 Structuring an SAF View of a Business Process

The process for applying the Survivability Analysis Framework requires the identification of a representative example(s), which must be decomposed into a series of steps. These steps must be a realistic view of how the business process actually happens for an as-is view or how it is expected to happen for a future perspective. The described actions must be consistent with what participants actually do. For each step, the required people, resources, and actions are assembled in a structured format to accurately portray who does each action, what initiates the action, what resources are critical to action performance, and the resulting outcomes.

To appropriately characterize a business process, it is imperative that all stakeholders agree with the information. The process of developing a well-articulated view of a business process that is shared by all stakeholders provides an opportunity to uncover differences in understanding, faulty assumptions, and ways in which organizational boundaries could contribute to stress and potential failure.

The remainder of this subsection of the report describes an example sequence of actions required to characterize a business process with the SAF. The full example of a business process spanning a doctor's office and hospital-associated lab is provided in Appendix A.

The survivability analysis starts by selecting an important business process and assembling a general description of what organizational need it addresses and why. In the doctor's office example, it is important that lab tests ordered for the patient be performed properly and results communicated to the doctor in a timely manner. Early diagnosis of critical patient conditions before they become crises is a goal for the physicians in this practice. Much of the diagnostic work is outsourced to local laboratories and hospitals. While patients may choose where to have tests performed, in many cases doctors are required to provide referrals. The Health Insurance Portability and Accountability Act (HIPAA) regulations control the sharing of patient identification data with the lab or hospital and their subsequent link to reporting of results back to the doctors. The selected cross-organizational business process example is as follows:

> A patient comes to the doctor for a follow-up visit. This individual was brought to the hospital emergency room several weeks prior with chest pains, treated for a mild heart attack, and released. The doctor, after examining the patient and reviewing the medical history along with the results of tests performed at the time of the office visit, orders further blood tests. Based on the results of these tests, a course of treatment is prescribed and communicated to the patient.

The sequence of actions required to perform this example can be described as follows:

A. Patient makes an appointment for an office visit to follow up on hospital release
B. Reminder sent to patient about scheduled office visit
C. Patient's available records are assembled for use in office visit
D. Patient arrives and checks in for scheduled appointment
E. Patient's insurance arrangements confirmed and co-payment made
F. Nurse moves office records and patient into examination room
G. Nurse takes vitals and electrocardiogram (EKG) (office policy for heart attack patients) and updates office hardcopy records in examination room for doctor
H. Doctor examines patient, reviews records and EKG
I. Doctor orders additional lab work
J. Hardcopy paperwork returned to medical records unit
K. Office visit information transcribed into office electronic medical record
L. Patient goes to lab for prescribed tests and registers at lab desk
M. Lab paperwork prepared and queued for phlebotomist

N. Phlebotomist takes blood, labels it for lab technician

O. Lab technician performs tests on sample and generates report

P. Lab results transmitted to hospital central repository

Q. Report transmitted to doctor's office (via email)

R. Doctor reviews test results, develops written treatment plan for patient (electronic or hardcopy)

S. Treatment plan communicated to patient

For each step in the example, a description of the preconditions, actions, and post conditions must be assembled. People and required resources must be identified. To assemble this view of the business process, additional information about the context in which the business process is performed and its participants is needed. The office context can be described as follows:

- Patient scheduling, electronic medical records, and billing are handled using a package system provided from the hospital (EPICARE), which includes the capability for authorized individuals to link to the hospital database and extract available patient data. The technical characteristics of this system are described in a manual from the hospital. The office has implemented it as a turn-key system with support provided (for a fee) by the hospital vendor.
- Everyone working at the doctor's office has individualized access to the system (nurses, doctors, office clerks, billing clerks, and office manager).
- Administrative control of the office system is handled by the medical records manager (also known as office manager).
- Technical support is provided electronically from the vendor (maintenance, troubleshooting, and upgrades).
- Everyone working at the office has been in their positions for several years.

The lab context is described as follows:

- LABTEST system is constructed to use the hospital database as an information repository and patient billing is handled by the hospital. The local office has applications for patient check-in, test paperwork management, results capture from test equipment, and doctor notification.
- Laboratory system activities are streamlined to handle large volumes of input.
- System development and support is handled by the lab group's central office.
- Local administrative support is provided through a contract with the local hospital in conjunction with the database connectivity.
- Staff turnover is high; few workers are in their positions beyond a year.

Using the available context information, each step in the business process example can be described. For step A the following table is constructed:

Table 2: SAF View of Example Step A

Step A	Patient makes an appointment for an office visit to follow up on hospital release
Preconditions	patient requires follow-up doctor's visit for hospital stay appointment staff has appropriate authorization to access scheduling, doctor availability, and patient demographic information telephone and computer system are available
Actions	patient calls doctor's office appointment staff answers phone appointment staff accesses, verifies, and updates patient contact information as needed appointment staff accesses doctor's schedule appointment date and time selected and updated with patient agreement appointment flagged as follow-up to hospital stay
Post conditions	appointment notification scheduled for day before appointment appointment is scheduled and in the system for proper patient, date, time, doctor

The description tables for the remainder of the steps in the medical example can be seen in Appendix A.

2.1.3 Identifying Critical Steps for Analysis

While it is possible to assemble a large amount of detailed information about each step in the process, this activity may not be useful. In order to guide the analysis, it is necessary to clearly articulate the goals of the business process. What constitutes successful business process completion? Many actions may be included which do not directly contribute to successful execution of the business process and would not warrant in-depth analysis. For this business process example, the following constitutes success:

- All ordered tests are appropriately performed in a timely manner and results accurately communicated to the requesting doctor.
- Patient information is transferred reliably and accurately in a timely manner with all privacy needs addressed.

A review of the steps critical to meeting the success criteria for the business process requires focused attention on steps L through Q. Of particular concern are steps O and P, where tests are performed and information is transferred from the lab to the doctor's office under the control of a third party (the hospital).

For each step selected for closer attention, we add claims describing how the actions in the step contribute to success of the business process. Step L is described as follows:

Table 3: SAF Critical Step View with Claims

Step L	Patient goes to lab for prescribed tests and registers at lab desk
Preconditions	patient has an order for lab work system is in place for collecting patient demographic and insurance information
Actions	collect patient insurance and billing information record doctor to receive report medical order entered into system
Post conditions	patient is queued for blood work medical order for lab work is properly entered into the system
Claims	all HIPAA privacy constraints are met patient information is accurately input into the laboratory system

The actions in this step are expected to support the goal for accuracy and privacy of patient information.

For steps of particular concern, potential causes of failure must be assembled to identify the ways in which completion of this step could be hampered (failure outcomes). For step O, the description would be expanded as follows:

Table 4: SAF Critical Step View with Failure Potentials

Step O	Lab technician performs tests on sample and generates report
Precondition	blood and paperwork ready technician loads proper machine with blood samples bar code on vial indicates patient and proper test to machine
Action	machine runs tests each machine sends results to lab's database collecting point results collated into report for transmission to the hospital repository
Post condition	report exists blood disposed of properly technician performing work is identified and linked to results
Claim	all required tests were run no unordered tests were run test results are accurately recorded test results are associated with the right patient lab audit trail exists—who did the work, who was the operator, and so forth access to results meets HIPAA requirements, such as technician cannot identify the patient associated with the test results

Step O	Lab technician performs tests on sample and generates report
Failure outcomes	missing (or delayed) results: • some or all tests are not done • some unrequested tests were performed wrong results: • results do not reflect the actual sample disclosure: • results are disclosed to unauthorized person • test results are not associated with the correct patient • test results are not associated with the correct doctor
Potential causes of failure	missing results • paperwork requiring tests to be run was lost or misplaced • blood samples were lost, contaminated, or misplaced • some tests were not run by the technician • wrong tests were run by the technician • some or all test results were not associated with the correct patient (in the lab) • some or all test results were not associated with the right doctor (in the lab) • lab database was inaccessible for receiving results • machine did not produce results • machine was not working and could not produce results wrong results • machine doing the test has an undetected internal failure so results were produced, but they are not the correct results • analysis machine is not calibrated, has faulty reagents, or similar faults disclosure • unauthorized entity (person, insurance company, or others) gained access to the analysis results during analysis (in the lab)

For convenience, the steps chosen for critical focus have been renumbered A1-A5. Appendix B has a full listing of the expanded description tables including the failure outcomes for steps A4 (O) and A5 (P).

Two summary views are constructed for the selected critical steps to focus attention on people and resources (key stress sources). The people view identifies each role involved in each step. If it is known, the controlling role (decision maker) for each step should be indicated so shifts in responsibility as well as organizational shifts can be visually articulated. These represent governance and policy change points where friction is likely. The table for steps A1-A3 appears as follows (controlling role is marked as "C" and participants are marked as "X"):

Table 5: SAF People Summary View for Steps A1 through A3

	A1) Patient to lab	A2) Lab prepares paper-work	A3) Blood sample drawn
Patient	X		X
Lab check-in staff	C	C	
Phlebotomist			C
Lab technician			

The resource table for these same steps is as follows:

Table 6: SAF Resource Summary View for Steps A1 through A3

	A1) Patient to lab	A2) Lab prepares paper work	A3) Blood sample drawn
Lab work order	X	X	
Patient insurance data	X		
HIPAA forms	X		
Lab scheduling	X	X	
Lab test repository and reporting system			
Blood sample			X
Lab paperwork (labels)		X	X
Testing machine			
Testing machine connectivity			
Doctor's office connectivity			

To better characterize more complex business processes, resources should be assembled in groups based on the way the organization has allocated management for them; resources controlled by a specific business unit would be grouped together. For example, resources controlled by the doctor's office would be grouped separately from those controlled by the laboratory or other third-party contracts. This provides visibility to potential variations in governance (policy) and allocation models (such as service level agreements) that could impact performance of the business process.

The full people and resource tables for steps A1-5 of the medical example are provided in Appendix B.

2.1.4 Evaluating Failure Potential

Stresses are the normal variations that occur constantly in the course of performing a business process. Some are expected variations that the process is constructed to accommodate such as higher volumes. In addition, unexpected errors and variations which the business process is not designed to accommodate can occur, leading to potential failure of a critical step and subsequent impact on the successful completion of the business process. In building the failure outcomes for each critical step as described in 2.1.3, a range of stress types and potential failures should be considered.

Large distributed systems are constructed incrementally. The functionality of the initial deployment of a system may suggest other applications that were not anticipated in the initial design. Users frequently exploit system functionality in unanticipated ways that improve the business processes but that may also stress the operation of components that were not designed for the new usage.

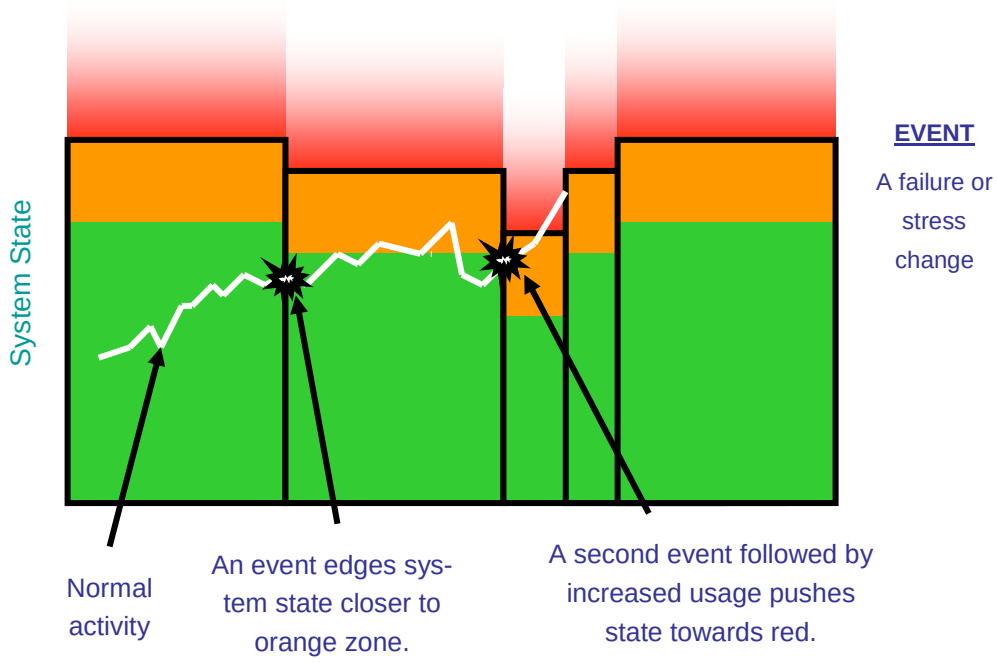


Figure 3: How Systems Fail

Business process failures can be caused by changes in usage as well as traditional causes such as hardware failures. Failures are frequently the result of multiple, often individually manageable errors that collectively become overwhelming. Using our medical example, a test equipment failure can delay test results for a significant number of patients. The delays temporarily reduce the available capacity to deal with other events. The occurrence of an additional problem such as transmission problems to the hospital database, combined with limited storage capacity for data at the lab, could lead to lost test results.

Figure 3 describes this pattern of system failure.

Stresses, when they exceed an expected range of tolerance, can also drive a business process into failure. Stresses may include

- interaction (data) triggered stress—missing, inconsistent, incorrect, unexpected, incomplete, unintelligible, out of date, duplicate
- resource triggered stress—insufficient, unavailable, excessive, latency, inappropriate, interrupted
- people-triggered stress—information overload, analysis paralysis, distraction (rubbernecking), selective focus (only looking for positive reinforcement), diffusion of responsibility (for example, "it's not my job"), lack of skills or training

Discrepancies (stresses and errors) arise normally in business processes. The overall success of a business process depends on how effectively discrepancies are accommodated through the people, resources, and actions that comprise the end-to-end process. Changes in business processes and systems can introduce new types of discrepancies. For example, a system that was developed for a local facility but is now supporting a national process for sharing information among many facilities may require revision to accommodate the increased complexity of information interchange. Dealing with discrepancies becomes much more difficult as the number of participants—people and systems—increases. Each participant has to deal with multiple sources of discrepancies, and a single discrepancy can affect multiple participants. There is increased likelihood that a poorly managed discrepancy will result in failures affecting additional participants.

- A business process breakdown results from a combination of failures that drive operational execution outside of acceptable limits.
- Work processes span multiple systems, and a failure of one system can affect the overall work process as well as other participating systems.
- Inconsistencies must be assumed as we compose systems:
  - Systems developed at different times exhibit variances in technology and expected usage.
  - A system will not be constructed from uniform parts; there are always some misfits, especially as the system is extended and repaired.

Human interactions may be necessary to bridge between systems, eroding the boundary between people and system and establishing critical business process dependencies on people interacting with multiple systems

2.2 VALUE PROVIDED BY USING A SHARED VIEW

The process of developing a well-articulated view of a business process that is shared by all stakeholders provides an opportunity to uncover differences in understanding, faulty assumptions, and ways in which organizational boundaries could contribute to stress and potential failure.

This is what SAF enables. An organization constructs a well-articulated view of example business processes documenting the interrelationships of people, process and technology. This shared view identifies critical steps and the ways in which these could fail, leading to business process failure.

Analysis of this information provides an opportunity to show how the various parts and pieces of technology fit (or should fit) together with the user and organizational aspects to form a repeatable and reliable end-to-end business process. Much of the information needed to assemble this view is scattered among a number of stakeholders and must be gathered through documents and workshops. Pilot usage of SAF has shown that most characterizations of business processes are idealized, providing insight into how they should work without considering what is actually in place. When technology is developed to only address ideal usage, actual operational usage is poorly supported. SAF provides a structure for gathering and visually assembling business process information that can be useful to management, users, technology architects, system engineers, and software engineers.

Many analysis techniques focus primarily on the technology systems and only consider people and resources in light of direct interactions with the technology. For SAF, the focus is on the end-to-end business process and is best initiated by consulting with those individuals responsible for actual execution of the business processes. Technology is only one component in the total business process and must be evaluated in light of its support of the key business drivers and operational success. The focus must be maintained on successful completion of the business process (satisfactory execution of each critical step). There are many ways in which a business process can be hampered. Those discrepancies that cause a critical impact to the end-to-end business process are the primary ones that warrant considerable investment in analysis.

Determining the critical steps and the failure outcomes can require the active participation of many stakeholders, including business process owners, functional and information subject matter experts, and operational resources knowledgeable in the organizational technology infrastructure. This brings together a range of knowledge that is usually broadly dispersed in the organization among people who have limited, if any, interaction. Though the steps to construct this shared view can be time consuming, drawing this dispersed information together in a shared view allows all organizational participants to understand their role in the process and the ways in which choices they make affect others.

The long-term value in assembling shared views of important business processes is the ability to consider the effect of change on operational success over time. With the availability of a shared view that includes the full range of people, process, and resource interactions, the impact of change can be expressed as its effect on the people, processes, and resources that make up the business process and contribute to its ongoing success. Proposed changes to a business process can be evaluated to determine potential problems for process success and requirements for effective mitigation.

In order to clearly articulate why and how critical steps in a business process are structured to effectively mitigate potential failure, an assurance case may be needed. The shared view of the business process constructed using SAF can be mined to provide much of the information needed in the development of an assurance case. This is the subject of the next section in this report.

3 Assurance Cases for Business Process Survivability

How do we establish confidence that a business process (including its underlying technology) is sufficiently survivable? Only by identifying all significant survivability threats[3] and showing why we believe they have been adequately controlled or eliminated. The Survivability Analysis Framework is an approach for identifying significant survivability threats. Assurance cases are an approach for showing why we believe that such threats[4] have been adequately addressed. Much like a legal case presented in a courtroom, a survivability assurance case is a comprehensive presentation of evidence, with argumentation linking the evidence to claims that important survivability properties have been assured. The evidence may consist of test results, formal analyses, simulation results, hazard analyses, modeling, inspections, and the like. The argument is the explanation of how the available evidence can reasonably be interpreted as indicating the required levels of survivability. Arguing survivability without evidence is unfounded. Evidence without supporting arguments is unexplained.

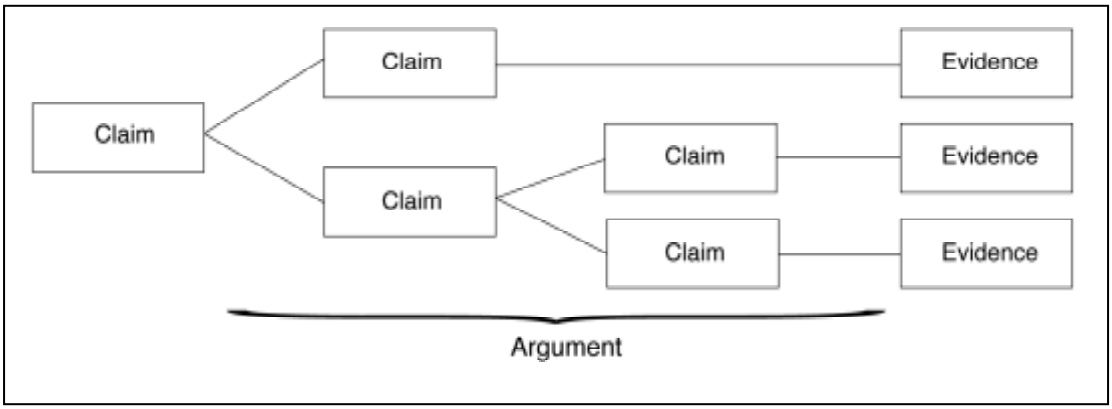


Figure 4: Claims, Arguments, Evidence

Figure 4 is a graphical representation of an assurance case, showing the relation between claims, argument, and evidence. The claim to the left is the overall claim. It is supported by two subclaims. The argument is that if those subclaims are valid then the overall claim is valid. The argument continues with further subclaims until, ultimately, a claim is supported by evidence, for which, by its very nature, no argument is necessary.

Assurance cases are not a new idea, and the notion is gaining interest in the United States [Jackson 2007]. In Europe (particularly) assurance cases have been used extensively to develop confidence that a system is acceptably safe. When used in this manner assurance cases are referred to

[3] A significant threat is one that presents a significant risk when combining the consequences of failure and the probability of failure.

[4] A threat to survivability is something done intentionally to exploit a vulnerability. A hazard is something that happens by accident. In this document we will use “threat” to mean either unless we explicitly state otherwise.

as *safety assurance cases*, or more simply, as a means of presenting a safety case. European laws require that safety cases be developed for (among other things) nuclear reactor systems, railway signaling systems, aircraft control systems, and other systems where safety is a significant concern. As safety cases have become more prevalent, notations, and tools supporting assurance case notations have been developed. One such notation is Goal Structuring Notation, or GSN. We use GSN in this section to document examples of survivability assurance cases. In the next subsection we describe GSN.

3.1 A NOTATION FOR ASSURANCE CASES

Creating and presenting assurance cases that are convincing to outside reviewers requires some care. Although such a case can be presented textually, a graphical representation can be much more easily understood by reviewers and more useful to the developers and maintainers of the system or business process being reviewed.

GSN is a graphical notation for showing the claims being made about a system and how arguments incorporating evidence support the claims [Kelly 2004]. Specific graphical symbols specify claims, evidence, argument strategy, and other elements of a case.

Figure 5 is a fragment of a generic assurance case. It shows a top-level *claim* ("The system under consideration is survivable") along with an argument structure supporting the claim. In GSN claims are stated as predicates (that is, true or false statements) and denoted by rectangles. The top level claim is annotated with an assumption about the claim (denoted by the oval with an "A" next to it), and some context information (denoted by a rectangle with round ends) that helps to explain the claim.

The argument is multi-pronged, as indicated by the *strategy* parallelogram, which is used to help the reviewer follow the argument. In this case, the argument strategy is to review each of the survivability hazards in turn, showing how each has been adequately mitigated (that is, controlled). One part of the argument, therefore, is the claim that "Hazard 1 is mitigated." Supporting evidence for this claim is shown in the evidence circle. Another part of the argument, consisting of the claim regarding hazard 2, is not fully developed in the fragment. This is denoted by the diamond under the rectangle. Finally, the third part of the argument claims that no other hazards significantly affect the survivability of the system. This, too, needs to be further developed. The overall argument is that if the subclaims ("Hazard 1 is mitigated" and "Hazard 2 is mitigated") are valid, and the claim that no other hazard has a significant effect on survivability is valid, then that is sufficient to show that the claim "The system under consideration is survivable" is valid.

There are other GSN symbols not shown in this example, such as an oval annotated with a "J" (a justification).

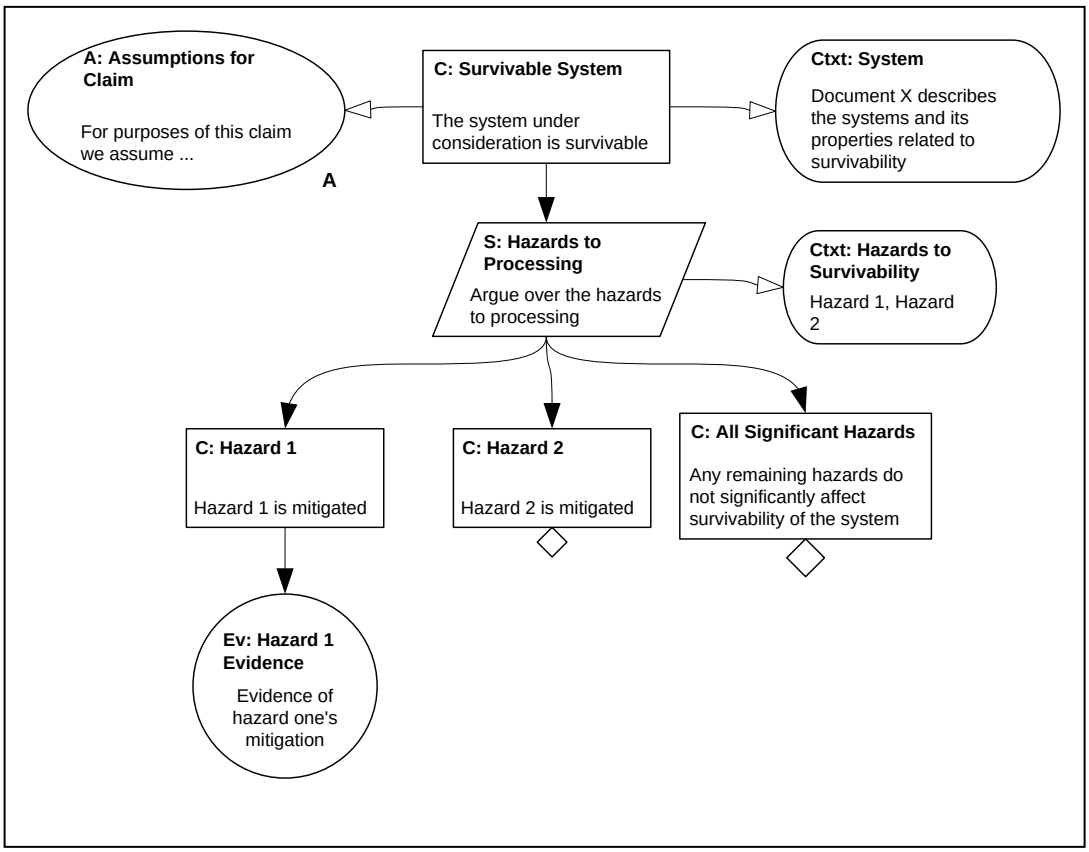


Figure 5: An Assurance Case Fragment

3.2 DEVELOPING AN ASSURANCE CASE

The first step in developing an assurance case is to determine broadly the structure of the case. This is not always easy because there are alternative ways of developing (and presenting) the argument. A well-chosen structure will make the assurance case easier to develop and review. One determinant of the structure of the assurance case is deciding what is to be assured. The classic safety case has only to show that the system remains in a safe state. It is not obligated to show that the system functions correctly when there are no current safety issues. Conversely, an assurance case designed to show that a system is survivable *must* worry about proper functionality in the presence of a threat or error condition. The optimal structure for a safety assurance case will likely not be the optimal structure for a survivability case.

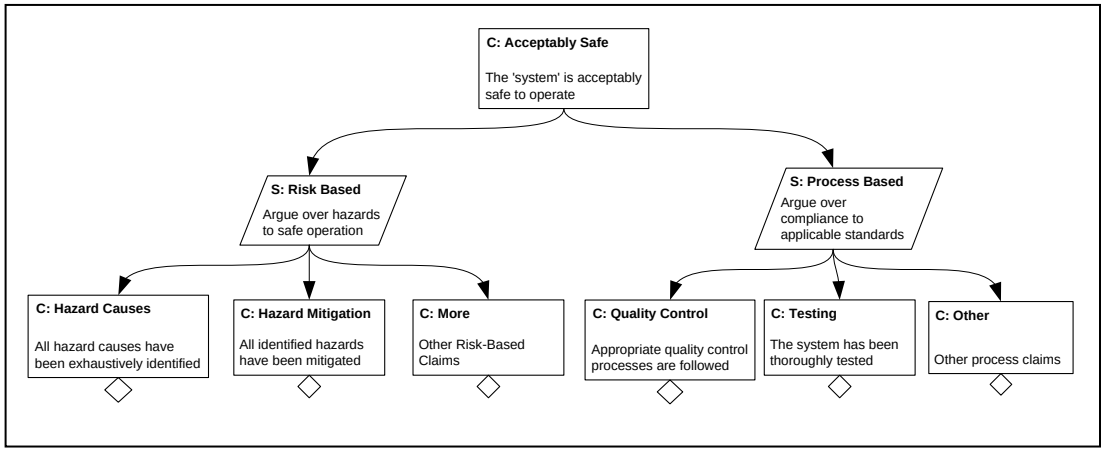


Figure 6: A Possible Top-Level Safety Case Structure

Figure 6 shows a possible top-level structure for a safety case. The argument is two pronged with one side evaluating risks specific to the system of interest and the other evaluating the competency of the developer. (The implied argument is that a competent developer is one who conforms to good development standards and a competent developer produces safe systems). Contrast this to a possible top-level structure for the survivability case shown in Figure 7. Notice that the claim labeled "processing" considers both hazards to be avoided and detecting problems when they inevitably occur. Breaking out the detection and mitigation from the actual hazard avoidance arguments results in, we think, a case that is easier to understand and review. An assurance case that is not understandable or reviewable does not provide a significant level of assurance, that is, confidence that the top-level claim is valid.

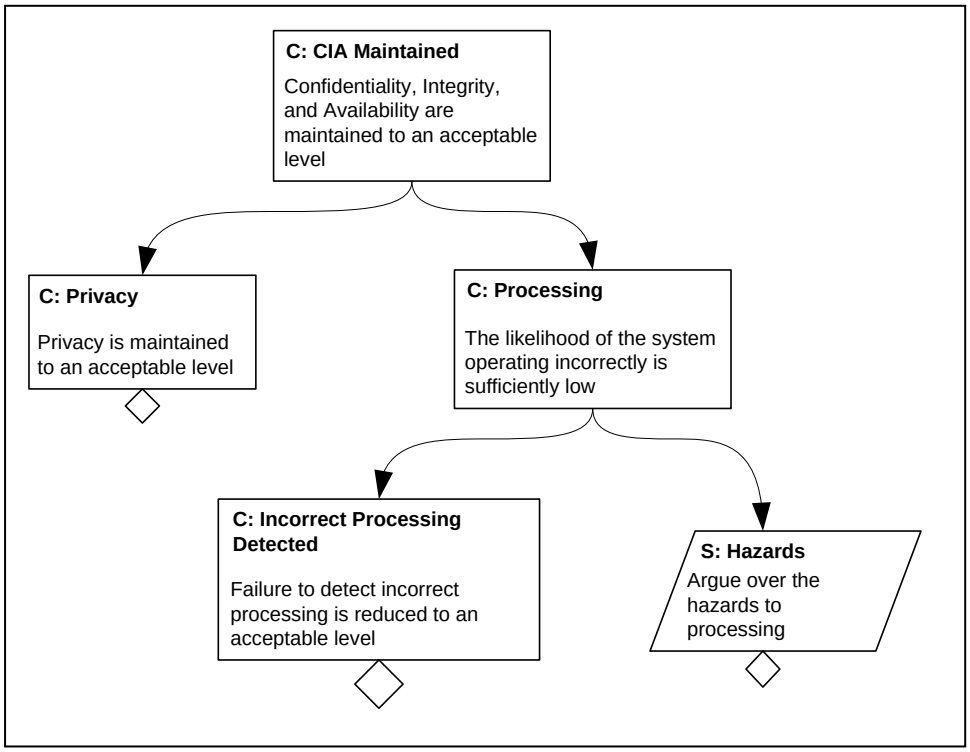


Figure 7: A Top-Level Survivability Case

Another determinant of the structure of an assurance case is the intended primary audience. People from different parts of an organization come with different agendas. For example, while an assurance case should be convincing to all, one that is security-centric in nature will be more convincing to some reviewers (that is, those with a security background) while one that is reliability-centric in nature will be more convincing to others—this in spite of the fact that both assurance cases purport to show the same top-level claim to be true.

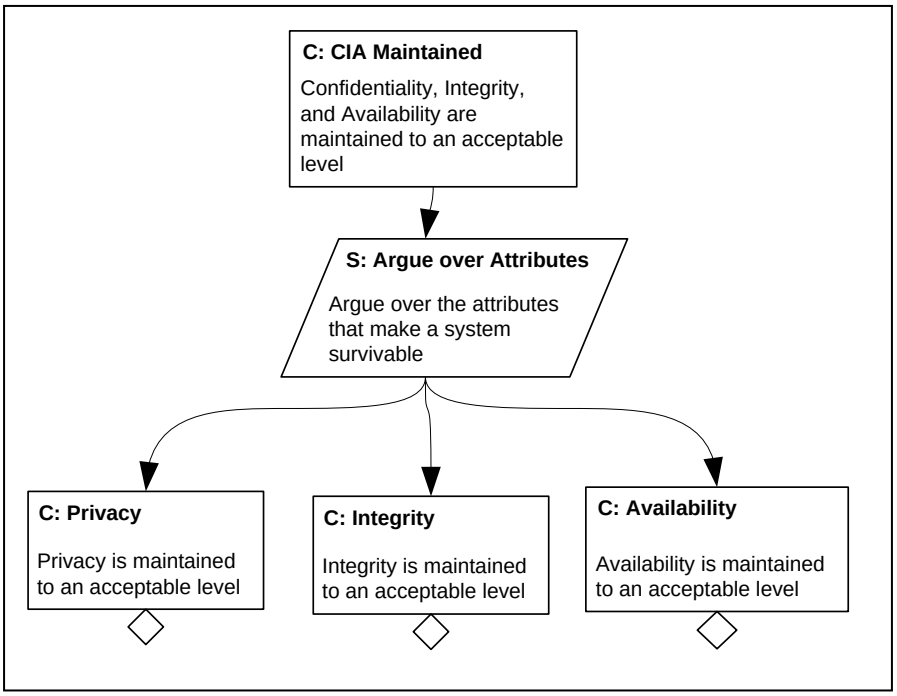


Figure 8: An Alternative Structure

Figure 8 shows another possible way to structure a survivability argument. Breaking the argument down by the constituent attributes of survivability is appealing—until you actually attempt to develop the arguments beneath each attribute. You soon find that there are interactions between the attributes and, perhaps worse, a lot of duplicate subarguments.

A well-structured assurance case captures a model of how you or the intended audience think about a problem—what is important, how things are related to each other, what needs to be highlighted. It structures the knowledge you have of a system and captures the implications of selected solutions to problems. It addresses issues that the target audience will find important, making it easier for them to find the important issues—and to see clearly how they have been resolved. Achieving all of this is hard to do. You can make several false starts before settling upon a structure that is comfortable.

3.3 A SURVIVABILITY ASSURANCE CASE

When being used to describe system survivability, an assurance case captures the full range of known relevant threats and vulnerabilities leading to significant failures. The linkage between threats and countermeasures is shown clearly by the claim, argument, evidence structure of an

assurance case. It's also easy to see weaknesses in countermeasures, as well as how those weaknesses have been addressed.

We'll illustrate this section with an assurance case covering step A4 in our example (reproduced below for convenience). Since the assurance case will be focused on how well confidentiality, integrity, and availability are maintained throughout the process of ordering, performing, and reporting results of blood tests, we have annotated the claims and failure outcomes with an indication of whether confidentiality, integrity, or availability is relevant.

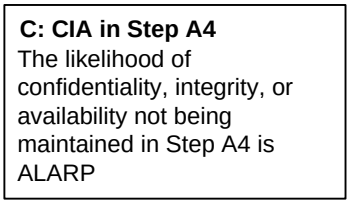
Step A4	Lab technician performs tests on sample and generates report
Preconditions	blood and paperwork ready technician loads proper machine with blood samples bar code on vial indicates patient and proper test to machine
Actions	machine runs tests each machine sends results to lab's database collecting point results collated into report for transmission to the hospital repository
Post conditions	report exists blood disposed of properly technician performing work is identified and linked to results
Claims	all required tests were run (integrity, availability) no unordered tests were run (integrity) test results are accurately recorded (integrity) test results are associated with the right patient (integrity) lab audit trail exists—who did the work, who was the operator, and so forth. (this claim is only needed to diagnose sources of failure, should a failure occur) access to results meets HIPAA requirements (for example, technician cannot identify the patient associated with the test results) (confidentiality)

Step A4	Lab technician performs tests on sample and generates report
Failure outcomes[5]	missing (or delayed) results: • some or all tests are not done (integrity, availability) • some unrequested tests were performed (integrity) wrong results: • results do not reflect the actual sample (integrity) disclosure: • results disclosed to unauthorized person (confidentiality) • test results not associated with the correct patient (integrity, confidentiality) • test results not associated with the correct doctor (integrity, confidentiality)
Potential causes of failure	missing results • paperwork requiring tests to be run was lost or misplaced (integrity) • blood samples were lost, contaminated, or misplaced (integrity) • some tests were not run by the technician (integrity) • wrong tests were run by the technician (integrity) • some or all test results were not associated with the correct patient (in the lab) (integrity, confidentiality) • some or all test results were not associated with the right doctor (in the lab) (integrity, confidentiality) • lab database was inaccessible for receiving results (availability) • machine did not produce results (availability) • machine was not working and could not produce results (availability) wrong results • machine doing the test has an undetected internal failure so results were produced, but they are not the correct results (integrity) • analysis machine is not calibrated or has faulty reagents, or similar faults (integrity) disclosure • unauthorized entity (person, insurance company, or others) gained access to the analysis results during analysis (in the lab) (confidentiality)

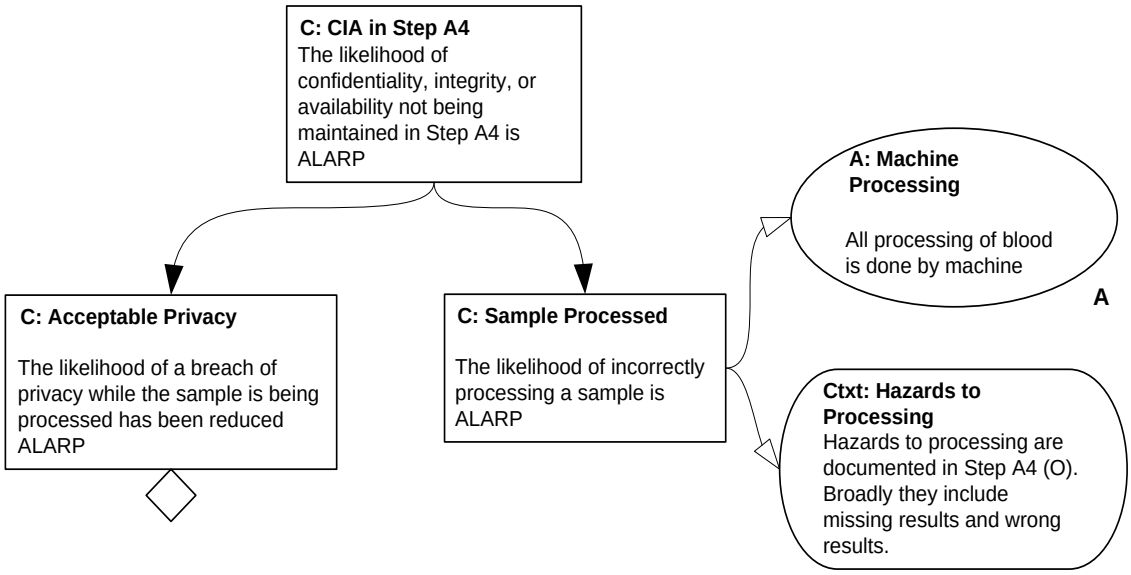
The first step in developing an assurance case for Step A4 is to formulate the main claim. In this case we want to show that confidentiality, integrity, and availability (CIA) properties are maintained during Step A4. The claim ends up being straightforward: "The likelihood of confidentiality, integrity and availability not being maintained is reduced as low as is reasonably practicable."

[5] We are interested in being explicit about failure outcomes because mitigating the causes and consequences of significant failure outcomes is the way to improve system survivability.

This claim has several noteworthy aspects. Perhaps the most important is that we are not claiming that "Step A4 processes blood correctly." This would be a so-called "sunny day" claim. Sunny day claims are difficult to work with because the argument beneath them tends to focus on functionality (that is, what needs to be done to process blood, in this case) rather than survivability (what hazards need to be avoided or mitigated). Instead we state the claim from a survivability viewpoint: we've reduced the possibility of hazards as low as is reasonably practicable. This brings us to the aspect worth noticing in the claim, namely the phrase "as low as is reasonably practicable." This phrase, introduced by safety case developers in the United Kingdom, is used frequently in assurance cases and is usually abbreviated as ALARP. ALARP means that the possibility of the hazard occurring has been reduced as low as makes sense given (1) the requirements of the system, and (2) the cost of reducing it further. In an ideal world, ALARP would mean that the hazard is completely eliminated, but in the real world there are some hazards that cannot be eliminated except at a cost out of proportion to the benefit obtained when considering the probability of the hazard occurring (e.g., "rarely occurs") and the impact of the failure if it does occur (e.g., "low impact"). ALARP allows the developers of the system to stop taking countermeasures once they've reduced the hazard "enough."



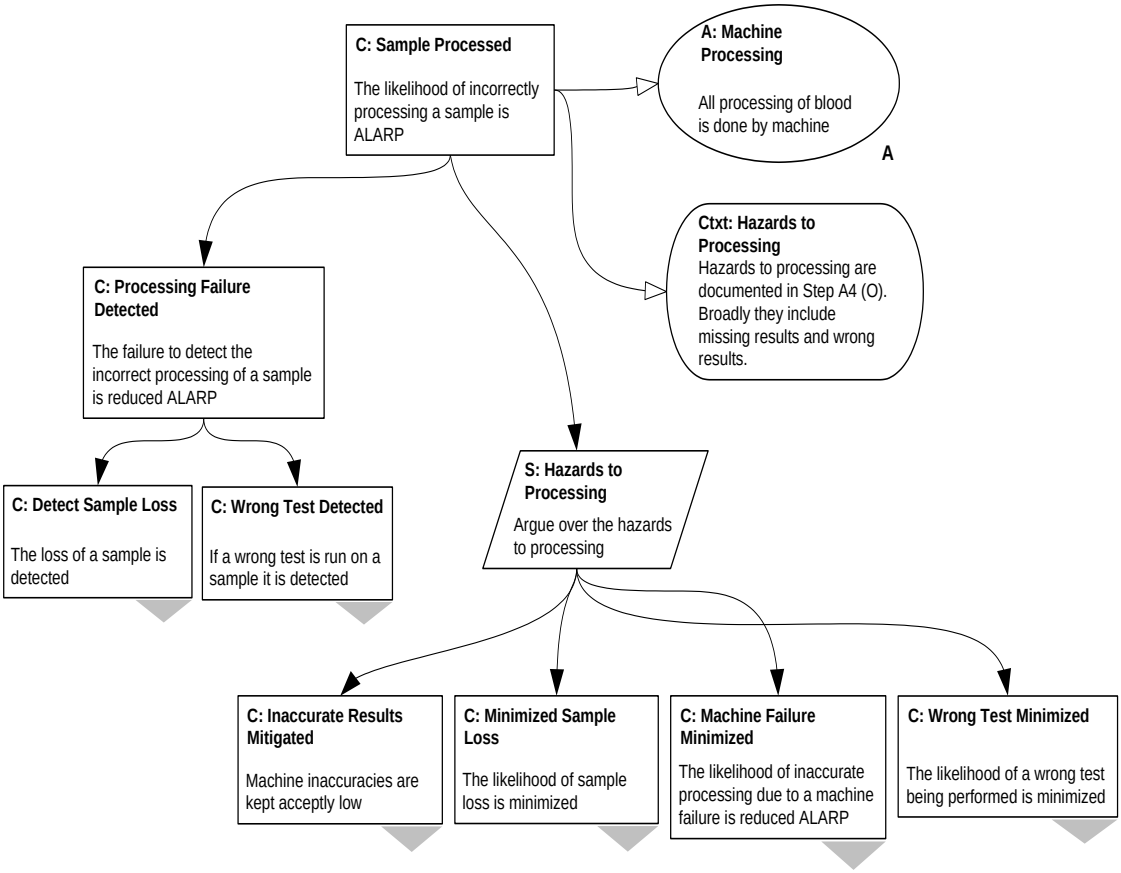
To expand this claim we note that there are three properties to show. We need to argue that confidentiality breaches are unlikely, that problems with system integrity are unlikely, and that system unavailability is unlikely.



The above shows this breakdown. Note, however, that we've elected to consider confidentiality ("Acceptable Privacy") separately from integrity and availability (covered under "Sample

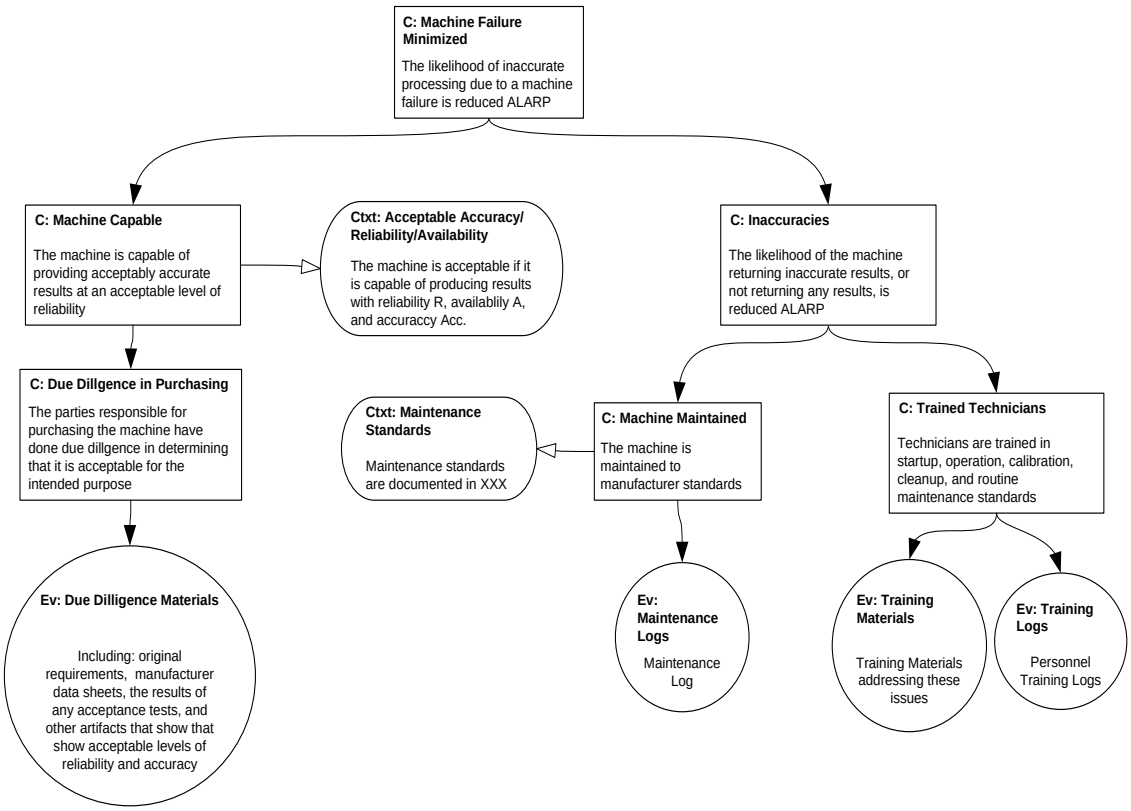
Processed"). The above fragment also shows an assumption, that the processing is all done by machine, and a context which refers to the hazards we'll be addressing. These elements help set expectations for the reader and reviewer. Without the assumption, for example, a reviewer of the expanded argument may wonder why we have left out some obvious human factors.

In this paper we'll concentrate on the "Sample Processed" side of the argument, leaving the "Acceptable Privacy" side to be expanded in the future.



To show that the likelihood of incorrectly processing a sample is reduced ALARP, we argue that hazards to normal processing are minimized and further that if one of the hazards actually causes a failure it will be detected (so that appropriate corrective actions can be undertaken or so that we can develop a record of how often particular hazards leads to a failure). The failures we need to detect are sample loss or that the wrong test was conducted on the sample. These are all illustrated above.

To continue our exposition we'll focus on one of the hazards, machine failure.



Notice that the argument to keep failures to a minimum largely depends upon purchasing a sufficiently reliable machine, maintaining it, and operating it correctly. Since the laboratory does not build the machine, it has no control over the machine other than at these points. The argument captures this.

The laboratory does have control over the training that its staff receives. If the training materials are weak then the arguments regarding hazards mitigated by a well-trained staff are correspondingly weaker. In general weak evidence or weak argument (such as an omitted claim or a specious link from a claim to a subclaim) leads to weak mitigation of hazards and a weaker overall assurance of survivability.

Notice that once the assurance case has been completely developed for a system, the evidence circles can be assembled into a check list. For a subsequent system development, as long as the assumptions within the assurance case remain valid and the new system is being developed in the same context as the old one, the check list can be used to determine if the new system will satisfy the same claim as the old one—without developing a new assurance case.

To summarize, in this subsection, we have shown how an assurance case can be used to develop increased confidence in the survivability of a business process. The assurance case discusses survivability hazards (or threats) that were identified as part of the SAF and shows how these hazards have been mitigated (that is, controlled). The case also shows what evidence needs to be collected to support the claims comprising the survivability argument. The intention is to provide a structure that is understandable to the variety of business process stakeholders.

4 Conclusion

SAF provides a structure for organizing information about a business process that incorporates people, process, and technology. SAF is most useful for environments where business processes rely on many independently constructed and supported systems, as described in Section 1.1 of this report.

The tables and graphical representations used in SAF and assurance cases structure the information collected for analysis and are readily used to highlight the strengths and gaps for survivability of a business process. As described in Section 3, this structure can be applied to the development of an assurance case to support claims about business process qualities such as security.

The steps required for applying SAF to a business process, described through an example in Section 2.1.2, are summarized as follows:

- Identify a representative example(s) of the business process.
- Decompose the example into the sequence of steps required for end-to-end execution of the business process.
- Describe the unique context of each organizational components involved in the business process.
- For each step construct a table of preconditions, actions, and post conditions that include all of the people, their roles, and resources needed to complete the step.
- Identify a subset of critical steps for further analysis.
- Assemble claims about the contribution of each selected step to overall mission success.
- Identify ways in which a step could fail to meet the specified claim in the form of failure outcomes and potential cause of these failures.
- Summarize people and resource usage across the critical steps to identify gaps in control and management of these key components.

The reader is encouraged to pilot the use of SAF with the selection of an important business process. The use of SAF to develop an example of a shared view for the selected example will support the following analyses:

- potential points of failure (stress analysis)
- survivability gaps (step interactions)
- mitigation strategies for critical business process failures
- identification of gaps in current people and resource requirements

Appendix A Example SAF Business Process

BUSINESS PROCESS EXAMPLE

A patient comes to the doctor for a follow-up visit. This individual was brought to the hospital emergency room several weeks prior with chest pains, treated for a mild heart attack, and released. The doctor, after examining the patient and reviewing the medical history along with the results of tests performed at the time of the office visit, orders further blood tests. Based on the results of these tests, a course of treatment is prescribed and communicated to the patient.

BUSINESS PROCESS STEPS

A. Patient makes an appointment for an office visit to follow up on hospital release

B. Reminder sent to patient about scheduled office visit

C. Patient's available records are assembled for use in office visit

D. Patient arrives and checks in for scheduled appointment

E. Patient's insurance arrangements confirmed and co-payment made

F. Nurse moves office records and patient into examination room

G. Nurse takes vitals and electrocardiogram (EKG) (office policy for heart attack patients) and updates office hardcopy records in examination room for doctor

H. Doctor examines patient, reviews records and EKG

I. Doctor orders additional lab work

J. Hardcopy paperwork returned to medical records unit

K. Office visit information transcribed into office electronic medical record

L. Patient goes to lab for prescribed tests and registers at lab desk

M. Lab paperwork prepared and queued for phlebotomist

N. Phlebotomist takes blood, labels it for lab technician

O. Lab technician performs tests on sample and generates report

P. Lab results transmitted to hospital central repository

Q. Report transmitted to doctor's office (email)

R. Doctor reviews test results, develops treatment plan for patient

S. Treatment plan communicated to patient

BUSINESS PROCESS CONTEXT

The office context can be described as follows:

- Patient scheduling, electronic medical records, and billing are handled using a package system provided from the hospital (EPICARE), which includes the capability for authorized individuals to link to the hospital database and extract available patient data. The technical characteristics of this system are described in a manual from the hospital. The office has implemented it as a turn-key system with support provided (for a fee) by the hospital vendor.
- Everyone working at the doctor's office has individualized access to the system (nurses, doctors, office clerks, billing clerks, and office manager).
- Administrative control of the office system is handled by the medical records manager (also known as office manager).
- Technical support is provided electronically from the vendor (maintenance, troubleshooting, and upgrades).
- Everyone working at the office has been in their positions for several years.

The lab context is described as follows:

- LABTEST system is constructed to use the hospital database as an information repository and patient billing is handled by the hospital. The local office has applications for patient check-in, test paperwork management, results capture from test equipment, and doctor notification.
- Laboratory system activities are streamlined to handle large volumes of input.
- System development and support is handled by the lab group central office.
- Local administrative support is provided through a contract with the local hospital in conjunction with the database connectivity.
- Staff turnover is high; few workers are in their positions beyond a year.

SAF STEP DESCRIPTIONS

Each step is described as to preconditions, actions, and post condition to fully characterize the interaction of people, process, and technology that must occur in order to complete each step.

Step A	Patient makes an appointment for an office visit to follow up on hospital release
Preconditions	patient requires follow up doctor's visit for hospital stay appointment staff has appropriate authorization to scheduling, doctor availability, and patient demographic information telephone and computer system are available
Actions	patient calls doctor's office appointment staff answers phone appointment staff accesses, verifies, and updates patient contact information

Step A	Patient makes an appointment for an office visit to follow up on hospital release
	as needed appointment staff accesses doctor's schedule appointment date and time selected and updated with patient agreement appointment flagged as follow up to hospital stay
Post conditions	appointment notification scheduled for day before appointment appointment is scheduled and in the system for proper patient, date, time, doctor

Step B	Reminder sent to patient about scheduled office visit
Preconditions	appointment scheduled for next day valid patient phone number available to scheduling system recorded message set up for appointment reminder service
Actions	scheduling system dials contact number and sends recorded message linked to appointment date and time
Post conditions	call is made to number on file with the appropriate information

Step C	Patient's available records are assembled for use in office visit
Preconditions	patient scheduled for appointment on current date appointment flagged as hospital visit follow up Medical Records has access to hospital patient records
Actions	matching of patient to proper records electronic and paper files (some identifier) office files pulled for use hospital data (discharge summary) extracted from hospital database into office electronic record and printed
Post conditions	updated office electronic record with hospital information updated hard copy for office visit use

Step D	Patient arrives and checks in for scheduled appointment
Preconditions	patient office records ready at check-in desk patient scheduled for appointment on current date doctor has not had emergency requiring schedule adjustments check in access to scheduling system

Step D	Patient arrives and checks in for scheduled appointment
Actions	patient match to office record file flag patient as checked in verify patient demographic data patient given HIPAA form to sign
Post conditions	signed HIPAA form patient sent to financial window with HIPAA form patient file queued for nurse pickup

Step E	Patient's insurance arrangements confirmed and co-payment made
Preconditions	patient standing at finance window patient has valid insurance card co-pay required (optional) access to scheduling system and patient electronic record access to insurer's data about the patient coverage
Actions	validate insurance information in patient electronic record co-pay collected (if required) and scheduling system tagged with payment
Post conditions	validated insurance information for patient patient registered for appointment with co-pay (if required)

Step F	Nurse moves office records and patient into examination room
Preconditions	patient office records queued for nurse patient in waiting room examination room available
Actions	examination room prepared for office visit patient and records moved to examination room
Post conditions	patient prepared for examination appropriate records are moved with the patient

Step G(a)	Nurse takes vitals
Preconditions	equipment for blood pressure, temperature, and other vitals ready for use

Step G(a)	Nurse takes vitals
Actions	performs required actions for doctor examination preparation notes collected data in patient record notified doctor patient is ready for examination
Post conditions	patient hard copy records annotated, ready for doctor

Step G(b)	Nurse takes EKG
Preconditions	EKG equipment ready for use
Actions	performs required actions for doctor examination preparation notes collected data in patient record notified doctor patient is ready for examination
Post conditions	patient EKG ready for doctor

Step H	Doctor examines patient, reviews records and EKG
Preconditions	patient ready for examination EKG results available vitals information available
Actions	doctor identifies potential health concerns doctor identifies actions to be taken to address concerns
Post conditions	doctor has and reviews all available information for patient

Step I	Doctor orders additional lab work
Preconditions	doctor has completed review of all available information (vitals, EKG, hospital discharge, prior medical history, and other information)
Actions	doctor completes lab order form (blood tests) doctor updates patient records (hardcopy) noting lab orders
Post conditions	lab order form given to patient to fulfill patient released from appointment

Step J	Hardcopy paperwork returned to medical records unit

Preconditions	doctor has completed patient examination doctor's interaction with patient has been incorporated into patient file
Actions	patient file returned to medical records area and filed
Post conditions	patient hardcopy medical documents stored for future retrieval

Step K	Office visit information transcribed into office electronic medical record
Preconditions	patient hardcopy records returned to medical records unit patient electronic medical record available for update transcribing resource had electronic access to electronic and hardcopy of medical records
Actions	additions to hardcopy medical record typed into electronic patient record
Post conditions	electronic medical record contains all hardcopy patient data

Step L	Patient goes to lab for prescribed tests and registers at lab desk
Preconditions	patient has an order for lab work system in place for collecting patient demographic and insurance information
Actions	collect patient insurance and billing information record doctor to receive report medical order entered into system
Post conditions	patient is queued for blood work medical order for lab work is properly entered into the system

Step M	Lab paperwork prepared and queued for phlebotomist
Preconditions	blood specimen requirements for each requested test are appropriately characterized within the system
Actions	print labels and orders for phlebotomist
Post conditions	paperwork (labels) printed for blood sample

Step N	Phlebotomist takes blood, labels it for lab technician
Preconditions	printed paperwork (labels) and patient ready
Actions	blood sample taken
Post conditions	blood in properly labeled vials

Step O	Lab technician performs tests on sample and generates report
Preconditions	blood and paperwork ready technician loads proper machine with blood sample bar code on vial indicates patient and proper test to machine
Actions	machine runs tests each machine sends results to lab's database collecting point results collated into report for transmission to the hospital repository
Post conditions	report exists blood disposed of properly technician performing work is identified and linked to results

Step P	Lab results transmitted to hospital central repository
Preconditions	test result report is available in the lab repository can match the lab's patient ID with the hospital's patient ID hospital can authenticate the lab communications exist lab can authenticate hospital lab can provide authorized readers of the transmitted report if the request for tests came directly to them from the patient or doctor (not via the hospital).
Actions	results transmitted
Post conditions	laboratory associated with results in hospital repository

Step Q	Notification given to doctor's office (email)
Preconditions	tests completed report exists doctor's email is provided
Actions	email sent to doctor's office notifying results are available results placed in patient medical record
Post conditions	information notification received

Step R	Doctor reviews test results, develops treatment plan for patient

Step R	Doctor reviews test results, develops treatment plan for patient
Preconditions	tests completed and report available at hospital central repository doctor received email notification doctor's office is able to access and retrieve report (authentication, authorization, and connectivity) doctor has connectivity and access to electronic medical record
Actions	doctor reviews test report doctor reviews office electronic medical record
Post conditions	treatment plan for patient is prepared (written) plan is given to nurse to notify patient

Step S	Treatment plan communicated to patient
Preconditions	treatment plan for patient is completed nurse has received treatment plan from doctor patient contact information and mailing address is available to the nurse
Actions	nurse calls patient to communicate treatment plan and arrange for subsequent patient actions as required by the plan letter prepared with treatment plan and information from nurse/patient discussion and mailed to patient treatment plan report and copy of letter added to patient office medical record
Post conditions	patient is notified of treatment plan and future actions (verbal and written) office medical record is updated with treatment plan and patient communications

Appendix B Mission Steps for Assurance Case

SUCCESSFUL BUSINESS PROCESS COMPLETION CRITERIA

- All ordered tests are appropriately performed in a timely manner and results accurately communicated to the requesting doctor.
- Patient information is transferred reliably and accurately in a timely manner with all privacy needs addressed.

FOCUS STEPS FOR ASSURANCE

A1. Patient goes to lab for prescribed tests and registers at lab desk (L)

A2. Lab paperwork prepared and queued for phlebotomist (M)

A3. Phlebotomist takes blood, labels it for lab technician (N)

A4. Lab technician performs tests on sample and generates report (O)

A5. Lab results transmitted to hospital central repository (P)

A6. Notification given to doctor's office (email) (Q)

Step A1	Patient goes to lab for prescribed tests and registers at lab desk
Preconditions	patient has an order for lab work system in place for collecting patient demographic and insurance information
Actions	collect patient insurance and billing information record doctor to receive report medical order entered into system
Post conditions	patient is queued for blood work medical order for lab work is properly entered into the system
Claims	all HIPAA privacy constraints are met patient information is accurately input into the laboratory system

Step A2	Lab paperwork prepared and queued for phlebotomist
Preconditions	blood specimen requirements for each requested test is appropriately characterized within the system
Actions	print labels and orders for phlebotomist

Step A2	Lab paperwork prepared and queued for phlebotomist
Post conditions	paperwork (labels) printed for blood sample
Claims	labels are accurate and legible (all and only requested tests; correct patient ID information) for requested tests

Step A3	Phlebotomist takes blood, labels it for lab technician
Preconditions	printed paperwork (labels) and patient ready
Actions	blood sample taken
Post conditions	blood in properly labeled vials
Claims	(none)

Step A4	Lab technician performs tests on sample and generates report
Preconditions	blood and paperwork ready technician loads proper machine with blood samples bar code on vial indicates patient and proper test to machine
Actions	machine runs tests each machine sends results to lab's database collecting point results collated into report for transmission to the hospital repository
Post conditions	report exists blood properly disposed of technician performing work is identified and linked to results
Claims	all required tests were run (integrity, availability) no unordered tests were run (integrity) test results are accurately recorded (integrity) test results are associated with the right patient (integrity) lab audit trail exists—who did the work, who was the operator, similar information access to results meets HIPAA requirements (for example, technician cannot identify the patient associated with the test results) (confidentiality)

Step A4	Lab technician performs tests on sample and generates report
Failure outcomes	missing (or delayed) results: • some or all tests are not done (integrity, availability) • some unrequested tests were performed (integrity) wrong results: • results do not reflect the actual sample (integrity) disclosure • results disclosed to unauthorized person (confidentiality) • test results not associated with the correct patient (integrity, confidentiality) • test results not associated with the correct doctor (integrity, confidentiality)
Potential causes of failure	missing results • paperwork requiring tests to be run was lost or misplaced (integrity) • blood samples were lost, contaminated, or misplaced (integrity) • some tests were not run by the technician (integrity) • wrong tests were run by the technician (integrity) • some or all test results were not associated with the correct patient (in the lab) (integrity, confidentiality) • some or all test results were not associated with the right doctor (in the lab) (integrity, confidentiality) • lab database was inaccessible for receiving results (availability) • machine did not produce results (availability) • machine was not working and could not produce results (availability) wrong results • machine doing the test has an undetected internal failure so results were produced, but they are not the correct results (integrity) • analysis machine is not calibrated, has faulty reagents, or similar faults (integrity) disclosure • unauthorized entity (person, insurance company, or others) gained access to the analysis results during analysis (in the lab) (confidentiality)

Step A5	Lab results transmitted to hospital central repository
Preconditions	test result report is available in the lab database can match the lab's patient ID with the hospital's patient ID hospital can authenticate the lab communications exist lab can authenticate hospital lab can provide authorized readers of the transmitted report if the request for tests came directly to them from the patient or doctor (not via the hospital).
Actions	results transmitted
Post conditions	laboratory associated with results in hospital repository
Claims	results transmitted without loss of patient privacy test results are accurately transmitted and entered correctly into the hospital database. test results are connected to the right patient test results are connected to the right doctor test results are connected to the right lab at the hospital (trusted sender) receipt of test results is acknowledged (non-repudiation) access to results in hospital repository meets HIPAA requirements an audit trail exists
Failure outcomes	missing (or delayed) results • some or all tests are reported missing from the database when they should have been present wrong results • results do not reflect the actual sample or doctor orders disclosure • results disclosed to unauthorized person • unauthorized person gains access to analysis results at the lab's database hospital system corrupted • what was transmitted (or the transmission process) causes a failure within the hospital information system duplicated results • entry of test results in hospital database duplicated

Step A5	Lab results transmitted to hospital central repository
Potential causes of failure	wrong results • test results are not applied to proper patient record in the repository • lab results modified before transmission (whether this is possible depends on the process of collecting results and then transmitting them) • results of tests not ordered are entered • test results were not accurately transmitted to the database (integrity) • transmitted results are tampered with (removed or changed) (integrity issue) • mismatch in hospital and lab data schema—likely a change on one end or the other missing results • authentication of lab fails (e.g. key mismatch) so results are not sent—could be critical. • test results expected but not received—lab loses results or does not transmit them (lab error recovery fails on a failure in its system) • tests were run but results were not made available to the database (results not transmitted or not received) • test results were not written to the database and no error message was received (or if received, results were not retransmitted) • data cannot be accessed by the hospital—encryption failure. • mismatch in hospital and lab data schema—likely a change on one end or the other disclosure • misconfigurations lead to lab system compromises • an unauthorized person has access to test results after the transmission (e.g., faxed to wrong number) • results are not associated with the correct doctor hospital system corrupted • poorly formed data record causes hospital system failures. • malware received from the laboratory duplicated results • retransmission (due to recovery from partial transmission) causes duplicate results to be entered in repository

Step A6	**Notification given to doctor's office (email)**
Preconditions	tests completed and loaded to the hospital database doctor's email is provided
Actions	email sent to doctor notifying that test results are available
Post conditions	information notification received at doctor's office

Step A6	**Notification given to doctor's office (email)**
Claims	notification sent to an accurate email address right doctor received patient notification no unauthorized person received notification notification contents are sufficient to properly identify the patient with no patient sensitive information

PEOPLE REFERENCE TABLE

	A1) Patient to lab	A2) Lab prepares paperwork	A3) Blood sample drawn	A4) Lab sample analyzed	A5) Report transmitted to hospital	A6) Notice sent doctor's office
Patient	X		X			
Lab check-in staff	C	C				
Phlebotomist			C			
Lab technician				C	C	

RESOURCE REFERENCE TABLE:

	A1) Patient to lab	A2) Lab prepares paperwork	A3) Blood sample drawn	A4) Lab sample analyzed	A5) Report transmitted to hospital	A6) Notice sent to doctor's office
Lab work order	X	X				
Patient insurance data	X					
HIPAA forms	X					
Lab scheduling	X	X				
Lab test repository and reporting system				X		X
Blood sample			X	X		
Lab paperwork (labels)		X	X	X		
Testing machine				X		
Testing machine connectivity				X		
Doctor office connectivity						X

References/Bibliography

URLs are valid as of the publication date of this document.

[Creel 2008]
Creel, Rita & Ellison, Robert J. *System-of-Systems Influences on Acquisition Strategy Development*. 2008.
https://buildsecurityin.us-cert.gov/daisy/bsi/articles/best-practices/acquisition.html

[Jackson 2007]
Jackson, Daniel; Thomas, Martyn; & Millett, Lynette I., editors. *Software for Dependable Systems: Sufficient Evidence*? The National Academies Press, 2007

[Kelly 2004]
Kelly, Tim & Weaver, Rob. *"The Goal Structuring Notation—A Safety Argument Notation."* http://www-users.cs.york.ac.uk/~rob/papers/DSN04.pdf (2004).

[Maier 1998]
Maier, Mark. "Architecting Principles for Systems of Systems." *Systems Engineering 1*, 4 (1998): 267-84.
http://www.infoed.com/Open/PAPERS/systems.htm

[Schwartz 2007]
Schwartz, John. "Who Needs Hackers?" *New York Times*, Sept. 12, 2007.

[US-Canada 2004]
U.S.-Canada Power System Outage Task Force. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. April 2004.
https://reports.energy.gov/

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	April 2008	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Survivability Assurance for System of Systems	FA8721-05-C-0003

6. AUTHOR(S)

Robert J. Ellison, John Goodenough, Charles Weinstock, Carol Woody

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2008-TR-008

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	ESC-TR-2008-008

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Complexity and change pervade today's organizations. Organizational and technology components that must work together may be created, managed, and maintained by different entities. Net-centric operations and service-oriented architectures will push this trend further, increasing the layers of people, processes, and systems. Existing analysis mechanisms do not provide a way to (1) focus on challenges arising from integrating multiple systems, (2) consider architecture tradeoffs carrying impacts beyond a single system, and (3) consider the linkage of technology to critical organizational functions. In response, a team at the Software Engineering Institute (SEI) built an analysis framework to evaluate the quality of the linkage among roles, dependencies, constraints, and risks for critical technology capabilities in the face of change.

The Survivability Analysis Framework (SAF), a structured view of people, process, and technology, was developed to help organizations analyze and understand stresses and gaps to survivability for operational and proposed business processes. The SAF is designed to

- identify potential problems with existing or near-term interoperations among components within today's network environments
- highlight the impact on survivability as constrained interoperation moves to more dynamic connectivity
- increase assurance that mission threads can survive in the presence of stress and possible failure

14. SUBJECT TERMS	15. NUMBER OF PAGES
Assurance, networked system, NSS, survivability, systems of systems	63

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18
298-102